

CS 2103

Assignment in Data Structures

(**insertPosition()**, **insertSorted()** and **delDuplicates()**)

Submitted by:

**Robert Michael Lim, BSCS – 2**

Submitted to:

**Ms. Christine F. Peña**

Given the structure definition of:

typedef struct node{

char elem;

struct node \*next;

}ctype, \*List;

**insertPosition()**

int insertPosition(List \*L, int pos, char x)

{

List \*trav, temp;

int ctr;

ctr = 1;

for(trav = L; \*trav!=NULL && ctr < pos; trav = &(\*trav)->next){

ctr++;

}

if(ctr == pos){

printf("%d\n\n", ctr);

temp = (List)malloc(sizeof(struct node));

if(temp!=NULL){

temp->elem = x;

temp->next = \*trav;

\*trav = temp;

}

ctr = 1;

}else{

ctr = 0;

}

return ctr;

}

**insertSorted()**

void insertSorted(List \*L, char x)

{

List \*trav, temp;

for(trav = L; \*trav!=NULL && (\*trav)->elem < x; trav = &(\*trav)->next){}

temp = (List)malloc(sizeof(struct node));

if(temp!=NULL){

temp->elem = x;

temp->next = \*trav;

\*trav = temp;

}

}

Given the structure definition of:

typedef struct{

char elem[SIZE];

int count;

}ArrayList;

**delDuplicates()**

void delDuplicates(ArrayList \*L, char elem)

{

int i, j;

for(i=0;i<L->count;i++){

for(j=i+1;j<L->count;){

if(L->elem[i]==L->elem[j]){

L->count--;

memcpy(L->elem+j, L->elem+j+1, sizeof(char)\*(L->count - j));

}else{

j++;

}

}

}

}